HN

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/707,625	11/07/2000	Donald C. Englin	RA 5266	9920

27516 7590 03/07/2005

UNISYS CORPORATION
MS 4773
PO BOX 64942
ST. PAUL, MN 55164-0942

EXAMINER	
CHU, GABRIEL L	
ART UNIT	PAPER NUMBER
2114	

DATE MAILED: 03/07/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/707,625	ENGLIN ET AL.	
	Examiner	**Art Unit**	
	Gabriel L. Chu	2114	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 November 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3,5-20 and 22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3,5-20 and 22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

Claim Rejections - 35 USC § 102

1. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

2. Claims 1-3, 5-7, 11, 12, 14, 15, 17-20 are rejected under 35 U.S.C. 102(b) as being anticipated by US 5509119 to La Fetra. Referring to claim 1, La Fetra discloses retrieving a stored tag address from the tag memory in response to the requester submitting a memory access address (From line 57 of column 3, " In operation, the CPU address Tag 201 is asserted by a CPU. The CPU-Tag is fed into the Tag input 207 of the comparator and simultaneously fed into the index input 203 of the cache RAM 205. The index input is the address input of the cache RAM. The cache RAM then outputs the Cache-Tag and the ECC information 213 associated with the memory location in the cache RAM addressed by the index input. In this example, Cache-Tag 211 and its associated ECC 213 are presented on the outputs of the cache RAM and fed into the inputs 217 and 219 respectively of the Tag check and correct circuit 215."); performing a first comparison of only the memory access address to the stored tag address, without regard to any error correction code associated with the stored tag address, to determine whether the requested data is stored in the cache memory (From line 2 of column 5 (with emphasis), "A second comparator 401 is provided which receives the Cache-Tag / Cache-ECC pair from the cache RAM 205. Another input of the second comparator 401 receives the CPU address Tag 403 and a derived CPU-Tag ECC generated by an ECC generator 405. The ECC generator 405 takes the CPU-Tag and computes the appropriate ECC for that Tag in the same manner as is used to derive the Cache-ECC

stored in the cache memory. This ECC generator derives the CPU-Tag ECC in the same or less time than is required to access the Cache-Tag/Cache-ECC pair from the cache RAM. **The second comparator 401 compares the Cache-Tag to the CPU-Tag and compares the derived CPU-Tag ECC to the Cache-ECC.** If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407. Since the ECC generator derives the CPU-Tag ECC in parallel with the cache RAM access, the time required to assert the fast hit signal is less than required to assert the true hit signal by the delay time through the Tag check and correction circuit 215 (approximately 2 ns)." Wherein the second comparator 401 disclosed by La Fetra is comprised of two comparisons, the Cache-Tag/CPU-Tag comparison and the Cache-ECC/CPU-ECC comparison. Further, from line 32 of column 4, "Given an address asserted by the CPU (CPU-Tag), the information that must be in the cache entry 101 for a cache hit to occur is predictable. That is, the Cache-Tag must match the CPU-Tag."); monitoring for errors in the stored tag address contemporaneously with the first comparison of the memory access address and the stored tag address (From line 1 of column 4, "The Tag check and correct circuit 215 checks the Cache-Tag for errors using the ECC information and corrects the Cache-Tag if an error is found. This checked and corrected Tag is then presented on the output 221 of the circuit and fed to the corrected Tag input 223 of the Tag comparator 209. After the corrected Tag is available to the Tag comparator 209, the comparator compares the corrected Tag to the CPU-Tag and if the two Tags match, the comparator outputs a true hit signal on output line 225. This signal is used by the cache memory system to provide data in the cache memory to the CPU by methods

known in the art. If the two Tags do not match, then there has been a cache miss and the cache is updated as previously discussed."); if a tag address error is detected in the stored tag address, disregarding the first comparison, correcting the tag address error, and performing a second comparison of the memory access address and the corrected tag address to determine whether the requested data is stored in the cache memory (From line 25 column 5, "If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before."); and if no tag address error is detected in the stored tag address, utilizing results of the first comparison of only the memory access address and the stored tag address to determine whether the requested data is stored in the cache memory (From line 22 column 5, "Since, most of the time the cache tag does not need to be corrected and generally there is a cache hit, by using the fast hit signal, the cache memory system can supply data to the CPU in less time than the prior art designs allow." From line 13 of column 5, "The second comparator 401 compares the Cache-Tag to the CPU-Tag and compares the derived CPU-Tag ECC to the Cache-ECC. If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407.").

3. Referring to claim 2, La Fetra discloses monitoring for errors in the stored tag address comprises identifying at least one error using a single error correction code associated with the stored tag address in the tag memory (From line 13 of column 5, "The second comparator 401 compares the Cache-Tag to the CPU-Tag and compares

the derived CPU-Tag ECC to the Cache-ECC. If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407.").

4. Referring to claim 3, La Fetra discloses the single error correction code is coded to provide error detection for the stored tag address and a plurality of configuration fields (From line 40 of column 3, "FIG. 1 illustrates a typical cache Tag-ECC entry 101 having three primary segments 103, 105 and 107. Segment 105 contains "house keeping" information such as bits that indicate the cache is "dirty" or "private". The ECC field 107 has error correcting data for both the Tag segment 103 and the house keeping segment 105.").

5. Referring to claim 5, La Fetra discloses the second comparison compares only the requested tag address with the memory access address, and disregards comparison of any stored error correction code bits (From line 1 of column 4, "The Tag check and correct circuit 215 checks the Cache-Tag for errors using the ECC information and corrects the Cache-Tag if an error is found. This checked and corrected Tag is then presented on the output 221 of the circuit and fed to the corrected Tag input 223 of the Tag comparator 209. After the corrected Tag is available to the Tag comparator 209, the comparator compares the corrected Tag to the CPU-Tag and if the two Tags match, the comparator outputs a true hit signal on output line 225.").

6. Referring to claim 6, La Fetra discloses performing the first comparison and monitoring for errors in the stored tag address occur contemporaneously with correcting the tag address error and performing the second comparison (From figure 4, the Fast Hit 407 and True Hit 225 are shown to operate in parallel. Further, from line 16 of

column 25, "Since the ECC generator derives the CPU-Tag ECC in parallel with the cache RAM access, the time required to assert the fast hit signal is less than required to assert the true hit signal by the delay time through the Tag check and correction circuit 215 (approximately 2 ns). Since, most of the time the cache tag does not need to be corrected and generally there is a cache hit, by using the fast hit signal, the cache memory system can supply data to the CPU in less time than the prior art designs allow. If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before.").

7. Referring to claim 7, La Fetra discloses correcting the tag address error and performing the second comparison are initiated upon recognition of a tag address error (From line 1 of column 4, "The Tag check and correct circuit 215 checks the Cache-Tag for errors using the ECC information and corrects the Cache-Tag if an error is found. This checked and corrected Tag is then presented on the output 221 of the circuit and fed to the corrected Tag input 223 of the Tag comparator 209. After the corrected Tag is available to the Tag comparator 209, the comparator compares the corrected Tag to the CPU-Tag and if the two Tags match, the comparator outputs a true hit signal on output line 225.").

8. Referring to claim 11, La Fetra discloses a cache hit detector, comprising: (a) a tag memory to store tag addresses corresponding to addresses currently cached (From figure 4, element 205); (b) a fast hit detection circuit, comprising: (i) a first address compare module coupled to the tag memory to receive a tag address and to compare only the tag address to a requested address (From line 2 of column 5, "A second

comparator 401 is provided which receives the Cache-Tag / Cache-ECC pair from the cache RAM 205. Another input of the second comparator 401 receives the CPU address Tag 403 and a derived CPU-Tag ECC generated by an ECC generator 405. The ECC generator 405 takes the CPU-Tag and computes the appropriate ECC for that Tag in the same manner as is used to derive the Cache-ECC stored in the cache memory. This ECC generator derives the CPU-Tag ECC in the same or less time than is required to access the Cache-Tag/Cache-ECC pair from the cache RAM. The second comparator 401 compares the Cache-Tag to the CPU-Tag and compares the derived CPU-Tag ECC to the Cache-ECC. If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407. Since the ECC generator derives the CPU-Tag ECC in parallel with the cache RAM access, the time required to assert the fast hit signal is less than required to assert the true hit signal by the delay time through the Tag check and correction circuit 215 (approximately 2 ns)."); (ii) an error detector coupled to the tag memory to receive the tag address and to determine whether there are any errors in the tag address (From line 2 of column 5, "A second comparator 401 is provided which receives the Cache-Tag / Cache-ECC pair from the cache RAM 205. Another input of the second comparator 401 receives the CPU address Tag 403 and a derived CPU-Tag ECC generated by an ECC generator 405. The ECC generator 405 takes the CPU-Tag and computes the appropriate ECC for that Tag in the same manner as is used to derive the Cache-ECC stored in the cache memory. This ECC generator derives the CPU-Tag ECC in the same or less time than is required to access the Cache-Tag/Cache-ECC pair from the cache RAM. The second comparator 401

compares the Cache-Tag to the CPU-Tag and compares the derived CPU-Tag ECC to the Cache-ECC. If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407. Since the ECC generator derives the CPU-Tag ECC in parallel with the cache RAM access, the time required to assert the fast hit signal is less than required to assert the true hit signal by the delay time through the Tag check and correction circuit 215 (approximately 2 ns)."); (iii) a gated output module coupled to the first address compare module and the error detector to output a fast hit indication if the requested address is stored in the tag memory as determined by the comparison of only the tag address and the requested address, and if and only if no error is discovered by the error detector and the requested address is stored in the tag memory (From line 2 of column 5 (with emphasis), "A second comparator 401 is provided which receives the Cache-Tag / Cache-ECC pair from the cache RAM 205. Another input of the second comparator 401 receives the CPU address Tag 403 and a derived CPU-Tag ECC generated by an ECC generator 405. The ECC generator 405 takes the CPU-Tag and computes the appropriate ECC for that Tag in the same manner as is used to derive the Cache-ECC stored in the cache memory. This ECC generator derives the CPU-Tag ECC in the same or less time than is required to access the Cache-Tag/Cache-ECC pair from the cache RAM. **The second comparator 401 compares the Cache-Tag to the CPU-Tag and compares the derived CPU-Tag ECC to the Cache-ECC.** If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407." Further, line 25 of column, "If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true

hit/miss signal as before."); (c) a slow hit detection circuit, comprising: (i) an error correction circuit coupled to the tag memory to receive the tag address and to correct errors in the tag address (From figure 4, element 215.); and (ii) a second address compare module coupled to the error correction circuit to receive the corrected tag address and to compare the corrected tag address to the requested address (From figure 4, element 209.).

9. Referring to claim 12, La Fetra discloses the fast hit detection circuit and the slow hit detection circuit are coupled in parallel such that the first address compare module and the error detector perform operations concurrently with operations of the second address compare module and the error correction circuit (From figure 4, the Fast Hit 407 and True Hit 225 are shown to operate in parallel. Further, from line 16 of column 25, "Since the ECC generator derives the CPU-Tag ECC in parallel with the cache RAM access, the time required to assert the fast hit signal is less than required to assert the true hit signal by the delay time through the Tag check and correction circuit 215 (approximately 2 ns). Since, most of the time the cache tag does not need to be corrected and generally there is a cache hit, by using the fast hit signal, the cache memory system can supply data to the CPU in less time than the prior art designs allow. If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before.").

10. Referring to claim 14, La Fetra discloses the tag memory further stores an error correction code for each block of data stored in the tag memory, wherein each block of data is associated with a single error correction code, and the single error correction

code provides error correction capabilities for the stored tag address and a plurality of configuration fields (From line 40 of column 3, "FIG. 1 illustrates a typical cache Tag-ECC entry 101 having three primary segments 103, 105 and 107. Segment 105 contains "house keeping" information such as bits that indicate the cache is "dirty" or "private". The ECC field 107 has error correcting data for both the Tag segment 103 and the house keeping segment 105.").

11. Referring to claim 15, La Fetra discloses the first address compare module and the error detector are coupled in parallel to contemporaneously compare the tag address to a requested address and determine whether there are any errors in the tag address (From line 2 of column 5, "A second comparator 401 is provided which receives the Cache-Tag / Cache-ECC pair from the cache RAM 205. Another input of the second comparator 401 receives the CPU address Tag 403 and a derived CPU-Tag ECC generated by an ECC generator 405. The ECC generator 405 takes the CPU-Tag and computes the appropriate ECC for that Tag in the same manner as is used to derive the Cache-ECC stored in the cache memory. This ECC generator derives the CPU-Tag ECC in the same or less time than is required to access the Cache-Tag/Cache-ECC pair from the cache RAM. The second comparator 401 compares the Cache-Tag to the CPU-Tag and compares the derived CPU-Tag ECC to the Cache-ECC. If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407. Since the ECC generator derives the CPU-Tag ECC in parallel with the cache RAM access, the time required to assert the fast hit signal is less than required to assert the true hit signal by the delay time through the Tag check and correction circuit

215 (approximately 2 ns)." Wherein the second comparator 401 disclosed by La Fetra is comprised of two comparisons, the Cache-Tag/CPU-Tag comparison and the Cache-ECC/CPU-ECC comparison.).

12. Referring to claim 17, La Fetra discloses the error detector determines whether there are any single bit errors in the tag address (From line 2 of column 5, "A second comparator 401 is provided which receives the Cache-Tag / Cache-ECC pair from the cache RAM 205. Another input of the second comparator 401 receives the CPU address Tag 403 and a derived CPU-Tag ECC generated by an ECC generator 405. The ECC generator 405 takes the CPU-Tag and computes the appropriate ECC for that Tag in the same manner as is used to derive the Cache-ECC stored in the cache memory. This ECC generator derives the CPU-Tag ECC in the same or less time than is required to access the Cache-Tag/Cache-ECC pair from the cache RAM. The second comparator 401 compares the Cache-Tag to the CPU-Tag and compares the derived CPU-Tag ECC to the Cache-ECC. If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407. Since the ECC generator derives the CPU-Tag ECC in parallel with the cache RAM access, the time required to assert the fast hit signal is less than required to assert the true hit signal by the delay time through the Tag check and correction circuit 215 (approximately 2 ns)." Wherein any comparison that is not equal constitutes at least a single bit error.).

13. Referring to claim 18, La Fetra discloses a data processing system comprising: (a) a main memory module for storing data (From line 14 of column 1, "a main memory".); (b) at least one cache memory coupled to the main memory module to

cache at least a portion of the data stored in the main memory module (From line 20 of column 1, "a cache".); (c) at least one processing unit (From line 13 of column 1, "a central processing unit".) to process data and to control data access with the main memory module and the cache memory, the processing unit comprising: (1) a tag memory to store tag addresses corresponding to addresses currently cached (From figure 4, element 205.); (2) a fast hit detection circuit, comprising: (i) a first address compare module coupled to the tag memory to receive a tag address and to compare only the tag address to a requested address (From line 2 of column 5 (with emphasis), "A second comparator 401 is provided which receives the Cache-Tag / Cache-ECC pair from the cache RAM 205. Another input of the second comparator 401 receives the CPU address Tag 403 and a derived CPU-Tag ECC generated by an ECC generator 405. The ECC generator 405 takes the CPU-Tag and computes the appropriate ECC for that Tag in the same manner as is used to derive the Cache-ECC stored in the cache memory. This ECC generator derives the CPU-Tag ECC in the same or less time than is required to access the Cache-Tag/Cache-ECC pair from the cache RAM. **The second comparator 401 compares the Cache-Tag to the CPU-Tag and compares the derived CPU-Tag ECC to the Cache-ECC.** If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407. Since the ECC generator derives the CPU-Tag ECC in parallel with the cache RAM access, the time required to assert the fast hit signal is less than required to assert the true hit signal by the delay time through the Tag check and correction circuit 215 (approximately 2 ns)."); (ii) an error detector coupled to the tag memory to receive the tag address and to

determine whether there are any errors in the tag address (From line 2 of column 5, "A second comparator 401 is provided which receives the Cache-Tag / Cache-ECC pair from the cache RAM 205. Another input of the second comparator 401 receives the CPU address Tag 403 and a derived CPU-Tag ECC generated by an ECC generator 405. The ECC generator 405 takes the CPU-Tag and computes the appropriate ECC for that Tag in the same manner as is used to derive the Cache-ECC stored in the cache memory. This ECC generator derives the CPU-Tag ECC in the same or less time than is required to access the Cache-Tag/Cache-ECC pair from the cache RAM. The second comparator 401 compares the Cache-Tag to the CPU-Tag and compares the derived CPU-Tag ECC to the Cache-ECC. If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407. Since the ECC generator derives the CPU-Tag ECC in parallel with the cache RAM access, the time required to assert the fast hit signal is less than required to assert the true hit signal by the delay time through the Tag check and correction circuit 215 (approximately 2 ns)."); (iii) a gated output module coupled to the first address compare module and the error detector to output a fast hit indication if the requested address is stored in the tag memory as determined by the comparison of only the tag address and the requested address, and if and only if no error is discovered by the error detector and the requested address is stored in the tag memory (From line 2 of column 5, "A second comparator 401 is provided which receives the Cache-Tag / Cache-ECC pair from the cache RAM 205. Another input of the second comparator 401 receives the CPU address Tag 403 and a derived CPU-Tag ECC generated by an ECC generator 405. The ECC generator 405

takes the CPU-Tag and computes the appropriate ECC for that Tag in the same manner as is used to derive the Cache-ECC stored in the cache memory. This ECC generator derives the CPU-Tag ECC in the same or less time than is required to access the Cache-Tag/Cache-ECC pair from the cache RAM. The second comparator 401 compares the Cache-Tag to the CPU-Tag and compares the derived CPU-Tag ECC to the Cache-ECC. If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407." Further, line 25 of column, "If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before."); (3) a slow hit detection circuit, comprising: (i) an error correction circuit coupled to the tag memory to receive the tag address and to correct errors in the tag address (From figure 4, element 215.); and (ii) a second address compare module coupled to the error correction circuit to receive the corrected tag address and to compare the corrected tag address to the requested address (From figure 4, element 209.).

14. Referring to claim 19, La Fetra discloses the fast hit detection circuit and the slow hit detection circuit are configured in parallel such that the first address compare module and the error detector perform operations concurrently with operations of the second address compare module and the error correction circuit (From figure 4, the Fast Hit 407 and True Hit 225 are shown to operate in parallel. Further, from line 16 of column 25, "Since the ECC generator derives the CPU-Tag ECC in parallel with the cache RAM access, the time required to assert the fast hit signal is less than required to assert the true hit signal by the delay time through the Tag check and correction circuit 215

(approximately 2 ns). Since, most of the time the cache tag does not need to be corrected and generally there is a cache hit, by using the fast hit signal, the cache memory system can supply data to the CPU in less time than the prior art designs allow. If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before.").

15. Referring to claim 20, La Fetra discloses the tag memory further stores an error correction code associated with each block of data, wherein each block of data is associated with a single error correction code, and the single error correction code! provides error correction capabilities for the stored tag address and a plurality of configuration fields (From line 40 of column 3, "FIG. 1 illustrates a typical cache Tag-ECC entry 101 having three primary segments 103, 105 and 107. Segment 105 contains "house keeping" information such as bits that indicate the cache is "dirty" or "private". The ECC field 107 has error correcting data for both the Tag segment 103 and the house keeping segment 105.").

Claim Rejections - 35 USC § 103

16. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

17. Claims 8-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 5509119 to La Fetra as applied to claim 1 above. Referring to claims 8 and 9, La Fetra discloses disregarding the first comparison results of the first comparison through an

output gate (From line 25 column 5, "If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before."). Although La Fetra does not specifically disclose said disregarding comprises blocking passage of the results by providing an error signal to the result signal's output gate when a tag address error is detected and disabling an output of the output gate upon receipt of the error signal, blocking a signal on error is notoriously well known in the art. Examiner takes official notice for a logic gate. A person of ordinary skill in the art at the time of the invention would have been motivated to block a signal on error using a logic gate because, from line 25 of column 5, "If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before." Wherein this being computer logic, a person of ordinary skill in the art at the time of the invention would have been motivated to pass the output through a "gate" of some kind, even if not explicitly disclosed as such, because a gate is an electronic switch that is the elementary component of a digital circuit that produces an electrical output signal that represents a binary 1 or 0 and is related to the states of one or more input signals by an operation of Boolean logic and La Fetra has disclosed that a tag error is detected (an input signal that represents a state) that results in the disallowance of a fast hit assertion (an electrical output signal that represents a binary 1 or 0).

18. Referring to claim 10, La Fetra discloses enabling the tag address error to be corrected and the second comparison of the memory access and corrected tag addresses to be performed in response to the error signal (From line 22 column 5,

"Since, most of the time the cache tag does not need to be corrected and generally there is a cache hit, by using the fast hit signal, the cache memory system can supply data to the CPU in less time than the prior art designs allow.").

19. Claims 13 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 5509119 to La Fetra as applied to claim 11 above, in further view of "Pipelined Datapath" from ***Logic and Computer Design Fundamentals*** by Mano et al. Referring to claim 13, La Fetra discloses means to coordinate timing between the fast hit detection circuit and the slow hit detection circuit (From line 25 column 5, "If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before."). Although La Fetra does not specifically disclose latching as this means, storing for subsequent operation is notoriously well known in the art. An example of this is shown by Mano et al. on page 348. More specifically, referring to figure 7-23, it can be seen in the operand fetch stage, as an example, a register file read incurs a 3 ns delay and a mux selection incurs a 1 ns delay; registers between the stages store data for operation by the next stage. As can be seen from this first stage, there are two separate datapaths: one for A data and one for B data. A data does not have a mux selection and therefore does not incur an additional 1 ns delay, however, since the next stage in the pipeline cannot use the data until both A data and B data are ready (and hence, a 4 ns total delay for that stage of the pipeline), that data is stored in the pipeline platform (registers, which are composed of latches, extremely basic elements of computer logic). Then in the next stage, when the data is ready, the data is read out and a function is applied to it. A person of

ordinary skill in the art at the time of the invention would have been motivated to store a result for subsequent operation because, as was shown by Mano et al., there are unequal delays for the data prior to arriving at the operation (From line 5 of page 348, "as soon as all of the tasks in a particular stage are done, the conveyor belt can move forward so that the same tasks can be performed on the next items on the belt.") and so that a task may be broken down into stages (From line 8 of page 346, "we need to be concerned about the speed or rate at which the microoperations are performed." Further from paragraph 3 of page 347, "A conveyor belt moves components from stage to stage by proceeding forward periodically the length of one stage. Components and partially completed assemblies are stored in bins."). La Fetra would have been motivated to use latches to store a result for subsequent operation because signals arrive unequally (From line 25 of column 5, "If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before.") and memory access also works in cycles.

20. Referring to claim 16, La Fetra discloses first address comparison results and a resulting error indicator signal (From line 2 of column 5, "A second comparator 401 is provided which receives the Cache-Tag / Cache-ECC pair from the cache RAM 205. Another input of the second comparator 401 receives the CPU address Tag 403 and a derived CPU-Tag ECC generated by an ECC generator 405. The ECC generator 405 takes the CPU-Tag and computes the appropriate ECC for that Tag in the same manner as is used to derive the Cache-ECC stored in the cache memory. This ECC generator derives the CPU-Tag ECC in the same or less time than is required to access the

Cache-Tag/Cache-ECC pair from the cache RAM. The second comparator 401 compares the Cache-Tag to the CPU-Tag and compares the derived CPU-Tag ECC to the Cache-ECC."). Although La Fetra does not specifically disclose latching the comparison and error indicator signal, wherein the comparison results and the error indicator signal are not passed to the gated output until both the comparison results and the error indicator signal are available at the latches, and until simultaneously clocked to concurrently provide the comparison results and the error indicator signal to the gated output, storing for subsequent operation is notoriously well known in the art. An example of this is shown by Mano et al. on page 348. More specifically, referring to figure 7-23, it can be seen in the operand fetch stage, as an example, a register file read incurs a 3 ns delay and a mux selection incurs a 1 ns delay; registers between the stages store data for operation by the next stage. As can be seen from this first stage, there are two separate datapaths: one for A data and one for B data. A data does not have a mux selection and therefore does not incur an additional 1 ns delay, however, since the next stage in the pipeline cannot use the data until both A data and B data are ready (and hence, a 4 ns total delay for that stage of the pipeline), that data is stored in the pipeline platform (registers, which are composed of latches, extremely basic elements of computer logic). Then in the next stage, when the data is ready, the data is read out and a function is applied to it. A person of ordinary skill in the art at the time of the invention would have been motivated to store a result for subsequent operation because, as was shown by Mano et al., there are unequal delays for the data prior to arriving at the operation (From line 5 of page 348, "as soon as all of the tasks in a particular stage are

done, the conveyor belt can move forward so that the same tasks can be performed on the next items on the belt.") and so that a task may be broken down into stages (From line 8 of page 346, "we need to be concerned about the speed or rate at which the microoperations are performed." Further from paragraph 3 of page 347, "A conveyor belt moves components from stage to stage by proceeding forward periodically the length of one stage. Components and partially completed assemblies are stored in bins."). La Fetra would have been motivated to use latches to store a result for subsequent operation because signals arrive unequally (From line 25 of column 5, "If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before.") and memory access works in cycles. Further, La Fetra would have been motivated to store for subsequent comparison and to gate a signal because, from line 15 of column 5, "If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407.", wherein La Fetra has clearly shown that the comparator 401 output is contingent upon both the Tag and ECC comparisons and a correct output cannot occur until both Tag and ECC comparisons have been generated.

21. Claims 13 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 5509119 to La Fetra as applied to claim 11 above, in further view of US 4310853 to Madson. Referring to claim 13, La Fetra discloses the need to coordinate timing between the fast hit detection circuit and the slow hit detection circuit (From line 25 column 5, "If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal

as before."). Although La Fetra does not specifically disclose latching as this means, latching to coordinate timing in a system is well known in the art. An example of this is shown by Madson from line 4 of column 5, "Latches 107 and 108, and in fact the other latches 140, 144, and 146 in the figures, are utilized as is known in the art, for uniformity and coordination of timing throughout the system." A person of ordinary skill in the art at the time of the invention would have been motivated to use latches because the signals of La Fetra arrive unequally (From line 25 of column 5 of La Fetra, "If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before.") and from line 4 of column 5 of Madson, "Latches 107 and 108, and in fact the other latches 140, 144, and 146 in the figures, are utilized as is known in the art, for uniformity and coordination of timing throughout the system."

22. Referring to claim 16, La Fetra discloses first address comparison results and a resulting error indicator signal (From line 2 of column 5, "A second comparator 401 is provided which receives the Cache-Tag / Cache-ECC pair from the cache RAM 205. Another input of the second comparator 401 receives the CPU address Tag 403 and a derived CPU-Tag ECC generated by an ECC generator 405. The ECC generator 405 takes the CPU-Tag and computes the appropriate ECC for that Tag in the same manner as is used to derive the Cache-ECC stored in the cache memory. This ECC generator derives the CPU-Tag ECC in the same or less time than is required to access the Cache-Tag/Cache-ECC pair from the cache RAM. The second comparator 401 compares the Cache-Tag to the CPU-Tag and compares the derived CPU-Tag ECC to

the Cache-ECC."). La Fetra further discloses the need to coordinate timing between the fast hit detection circuit and the slow hit detection circuit (From line 25 column 5, "If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before."). Although La Fetra does not specifically disclose latching as this means, latching to coordinate timing in a system is well known in the art. An example of this is shown by Madson from line 4 of column 5, "Latches 107 and 108, and in fact the other latches 140, 144, and 146 in the figures, are utilized as is known in the art, for uniformity and coordination of timing throughout the system." Further referring to figure 1a of Madson, see clock. A person of ordinary skill in the art at the time of the invention would have been motivated to use latches because the signals of La Fetra arrive unequally (From line 25 of column 5 of La Fetra, "If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before.") and from line 4 of column 5 of Madson, "Latches 107 and 108, and in fact the other latches 140, 144, and 146 in the figures, are utilized as is known in the art, for uniformity and coordination of timing throughout the system."

23. Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over US 5509119 to La Fetra. Referring to claim 22, La Fetra discloses a cache hit detector, comprising: (a) means for storing tag memory addresses corresponding to addresses of data currently stored in cache memory (From figure 4, element 205.); (b) means for providing alternate cache hit detection via concurrent processing on each of at least two cache hit detection paths, the alternate cache hit detection means comprising: (1) first

hit detection path means for detecting cache hits without first performing error detection and correction (From figure 4, "Fast Hit" path 407.); (2) second hit detection path means for detecting cache hits, the second hit detection path means comprising: (i) means for detecting errors in the tag memory address (From figure 4, element 215.); (ii) means for correcting the tag memory address if errors in the tag memory address are discovered (From figure 4, element 215.); (iii) means for detecting for cache hits using the corrected tag memory address if errors in the tag memory address are discovered (From figure 4, element 209.); and (iv) means for disabling the first hit detection path means if errors in the tag memory address are discovered (From line 25 column 5, "If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before."); and (c) a need for a means for coordinating timing between the first hit detection path means and the second hit detection path means (From line 22 of column 5 (with emphasis), "Since, most of the time the cache tag does not need to be corrected and generally there is a cache hit, by using the fast hit signal, the cache memory system can supply data to the CPU in less time than the prior art designs allow. If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system **waits** for the true hit/miss signal as before."

As Attorney has astutely pointed out, what the means for timing coordination may be, under 112 6th, refers to a specific embodiment of Applicant's specification (although Examiner notes, no specific embodiment of this means has been pointed out by Applicant) and not any and every embodiment imaginable; Examiner refers to

Applicant's specification and determines an embodiment of this timing means to be a **clock**. Although La Fetra does not explicitly disclose that this means may be a clock, La Fetra does disclose from line 37 of column 2 (with emphasis), "With CPU **clock** speeds now in the 130 MHZ range, the time taken to check the cache address tag represents a significant time delay in the processing process. Also, since a good cache design yields a high percentage of "hits" and errors are infrequent, the cache address tag seldom has to be corrected and the delay time through the correction circuit unnecessarily penalizes almost every memory cycle." Further noting that La Fetra is teaching **a cache in a CPU** and that CPUs, particularly of the kind taught by La Fetra operate synchronously, with a CPU clock.). A person of ordinary skill in the art at the time of the invention would have been motivated to use La Fetra's own clock for timing because the signals of La Fetra's cache hit detection system are propagating at different speeds and would need to **wait** upon an ECC miscompare (From line 22 of column 5 (with emphasis), "Since, most of the time the cache tag does not need to be corrected and generally there is a cache hit, by using the fast hit signal, the cache memory system can supply data to the CPU in less time than the prior art designs allow. **If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system waits for the true hit/miss signal as before.**"), and only one of the fast hit and true hit/miss signals is used for the cache hit signal.

Response to Arguments

24. Applicant's arguments filed 15 November 2004 have been fully considered but they are not persuasive. Regarding Applicant's argument (page 8) that La Fetra does

not, as claimed, only compare the stored tag address with the requested address for the fast hit comparison, from line 13 of column 5 of La Fetra, "The second comparator 401 compares the Cache-Tag to the CPU-Tag and compares the derived CPU-Tag ECC to the Cache-ECC. If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407." Wherein, clearly, the comparison of ECC and address bits are treated separately. However, assuming arguendo that this is one comparison procedure for the entirety of bits, a person of ordinary skill in the computer logic arts would recognize that the address and the ECC are comprised of bits, and the comparison thereof of is on a bit by bit basis. Therefore, since one bit's comparison has nothing to do with another bit's comparison, the bits corresponding to the address are compared exclusively of the comparison of the bits corresponding to the ECC.

However Applicant brings up a pertinent point: the claim seemingly compares address data separate from error data, and the prior art, La Fetra seemingly compares address data with error data. Looking at the function of Applicant's invention however, it will become clear that error considerations are also used for the fast hit comparison. Simply put, "HIT INFO" is not a hit decision at all, but, as it is written, preliminary hit or miss information that has no value until the error state of the tag value is taken into consideration (see figure 3, the output of 316). This is precisely what La Fetra accomplishes in gate 401, i.e., for the comparisons that the comparators perform, positive comparison indicates coincidence of CPU-Tag and Cache-Tag. It is what these comparisons mean combined that generates a cache hit signal, From line 15 of La

Fetra, "If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407"

Applicant continues to obfuscate this issue with the argument (page 9) "Claim 1 requires no ECC comparisons to issue a fast cache hit result", when in fact, an ECC comparison is integral to the fast hit decision, again see figure 3, 310, 316.

25. Referring to Applicant's argument (page 10) regarding claim 2 and a "single" ECC, Examiner thanks Applicant for the proper definition of single and further points to the "single error correction code associated with the stored tag address in the tag memory" disclosed from line 13 of column 5 (with emphasis), "The second comparator 401 compares the Cache-Tag to the CPU-Tag and compares the derived CPU-Tag ECC to the **Cache-ECC**. If both the Tags and the ECCs match, then the comparator 401 outputs a fast hit signal 407."

26. Referring to Applicant's argument (page 11) regarding the absence of specific means in view of 112 6th paragraph for "means for coordinating timing", From line 22 of column 5 (with emphasis), "Since, most of the time the cache tag does not need to be corrected and generally there is a cache hit, by using the fast hit signal, the cache memory system can supply data to the CPU in less time than the prior art designs allow. If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system **waits** for the true hit/miss signal as before." As Attorney has astutely pointed out, what **this means** may be, and under 112 6th, means plus function language **refers to a specific embodiment of Applicant's specification** (although Examiner notes, no specific embodiment of this means has been pointed out

by Applicant) and not any and every embodiment imaginable, **Examiner** refers to Applicant's specification and **determines an embodiment of this timing means to be a clock**. Further, from line 37 of column 2 (with emphasis), "With CPU **clock** speeds now in the 130 MHZ range, the time taken to check the cache address tag represents a significant time delay in the processing process. Also, since a good cache design yields a high percentage of "hits" and errors are infrequent, the cache address tag seldom has to be corrected and the delay time through the correction circuit unnecessarily penalizes almost every memory cycle." Further noting that La Fetra is teaching a cache in a CPU and that CPUs, particularly of the kind taught by La Fetra operate synchronously, with a CPU clock.

27. Regarding Applicant's argument that the motivation to combine La Fetra and LCDFundamentals has not been established, Examiner apologizes if the motivation to combine timing circuits from fundamental logic and computer designs with a computer logic circuit that has timing issues is not clear. Stated herein, a person of ordinary skill in the art at the time of the invention would have been motivated to use a clock (and the trappings of multi-cyclical coordination such as memory comprising latches as shown) for timing because the signals of La Fetra are propagating at different speeds (From line 22 of column 5 (with emphasis), "Since, most of the time the cache tag does not need to be corrected and generally there is a cache hit, by using the fast hit signal, the cache memory system can supply data to the CPU in less time than the prior art designs allow. If the fast hit is not asserted, then either there is a cache miss or the Cache-Tag has an error and the cache memory system **waits** for the true hit/miss signal as before."), only

one of these signals is used for the cache hit signal, and a decision between the two must be made. Similarly, clocks are used in pipelines, as a fundamental design for computer logic, because signals in any given stage of a pipeline may not propagate at the same speed, and indeed, as shown, stages of data processing operate at different stages/rates, so a clock dictates when a signal must be settled so as to propagate the proper signal from one stage of a pipeline to the next.

Applicant further argues that "if the nature of the problem is to wait for a signal, a resistor-capacitor (RC) network could be used or any other number of circuits differing from the claimed language". Examiner notes that, as Attorney points out, alternate solutions for issues of logic timing are well known in the art, and are indeed fundamental to logic and computer design. If the need arises, this very point will be used if another "means for coordinating timing" is used in place of a clock.

Conclusion

28. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

US 5916314 to Berg et al.

US 6629206 to Johnson

US 6681299 to Shimamura et al., from line 8 of column 3, "When there is no error in TRUE-TAG#0 as a result of this checking, the error checking circuit 5 makes a hit decision by using TRUE-TAG#0 (step S13). Thereafter, TRUE-TAG#0 is directly input to the cache TAG-RAM 1 via the inverter 4, and the stored TRUE-TAG#0 and SHADOW-TAG#0 are updated (step S14). On the other hand, in the step S12, when

TRUE-TAG#0 is in error, SHADOW-TAG#0 is retrieved (step S15), and the error checking circuit 5 carries out error checking (step S16). In this case, when there is no error in SHADOW-TAG#0 as a result of this checking, the error checking circuit 5 makes a hit decision by using the data of SHADOW-TAG#0 (step S17)."

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Gabriel L. Chu whose telephone number is (571) 272-3656. The examiner can normally be reached on weekdays between 8:30 AM and 5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Robert W. Beausoliel, Jr. can be reached on (571) 272-3645. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Bryce P. Bonzo
Bryce P. Bonzo
Primary Examiner
AU 2114

gc